



# Travelling Cost

Link submit: <http://www.spoj.com/problems/TRVCOST/>

Solution:

C++	<a href="http://ideone.com/46vLwO">http://ideone.com/46vLwO</a>
Java	<a href="http://ideone.com/hIDBX9">http://ideone.com/hIDBX9</a>
Python	<a href="http://ideone.com/aA8Qa2">http://ideone.com/aA8Qa2</a>

Tóm tắt đề:

Chính phủ nơi Rohit sống chọn một số địa điểm trong thành phố để xây dựng đường đi và các địa điểm được đánh số từ 0, 1, 2, ..., 500. Chính phủ muốn làm những con đường giữa các cặp địa điểm khác nhau và quy định chi phí đi lại trên con đường đó là  $W$ .

Bạn hãy giúp Rohit tìm chi phí thấp nhất để đi từ địa điểm  $U$  đến  $Q$  địa điểm khác trong thành phố nhé.

Input:

Dòng đầu tiên chứa số nguyên  $N$  ( $1 \leq N \leq 500$ ) – số đường đi.

$N$  dòng tiếp theo, mỗi dòng chứa ba số nguyên  $A, B, W$  ( $0 \leq A, B \leq 500, 1 \leq W \leq 100$ ) tức đường đi từ  $A$  đến  $B$  có chi phí là  $W$ .

Dòng tiếp theo chứa số nguyên  $U$  là thành phố mà Rohit muốn tìm đường đi đến các nơi khác.

Dòng tiếp theo chứa số nguyên  $Q$  là số lượng thành phố cần tìm đường đi đến đó.

$Q$  dòng còn lại, mỗi dòng chứa một số nguyên  $V$  là thành phố cần đến.

Output:

Với mỗi số thành phố  $V$  trong input bạn hãy in ra chi phí ngắn nhất để đến thành phố này.

Ví dụ:

7	4
0 1 4	5
0 3 8	9
1 4 1	NO PATH
1 2 2	
4 2 3	

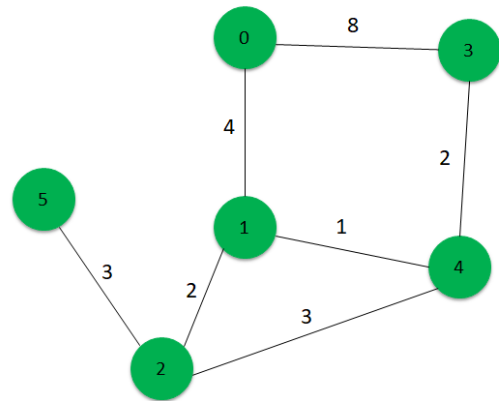
2	5	3
3	4	2
0		
4		
1		
4		
5		
7		

### Giải thích ví dụ:

Có 7 đường đi trong thành phố và được biểu diễn bằng đồ thị như hình bên.

Đường đi ngắn nhất từ:

- $0 \rightarrow 1$ : chi phí 4.
- $0 \rightarrow 4$ : chi phí 5.
- $0 \rightarrow 5$ : chi phí 9.
- $0 \rightarrow 7$ : không có đường đi.



### Hướng dẫn giải:

Đề cho biết các địa điểm được đánh số từ 0 đến 500 và danh sách các đường đi, vậy ta có thể thấy đây là một đồ thị có 501 đỉnh và các con đường sẽ là các cạnh của đồ thị.

Từ danh sách cạnh, tạo đồ thị, sau đó chạy Dijkstra từ đỉnh U. Sau khi chạy xong dựa vào mảng chi phí dist để tìm chi phí đến các đỉnh cần tìm. Nếu đỉnh nào chi phí là vô cực (INF) thì sẽ không có đường đi đến đó.

**Độ phức tạp:**  $O(E \log V)$  với E là số lượng cạnh (cung), V là số lượng đỉnh trong đồ thị.